Computational Geometry: Theory and Applications 3 (1993) 327–351
Elsevier

Representing polyhedra: faces are better than vertices

Lenwood S. Heath, Praveen K. Paripati and John W. Roach

Department of Computer Science, 562 McBryde Hall, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061–0106, USA

Communicated by David Dobkin
Submitted 30 January 1991
Accepted 23 August 1993

Abstract

This paper investigates the reconstruction of planar-faced polyhedra given their spherical dual representation. The spherical dual representation for any genus 0 polyhedron is shown to be unambiguous and to be uniquely reconstructible in polynomial time. It is also shown that when the degree of the spherical dual representation is at most four, the representation is unambiguous for polyhedra of any genus. The first result extends, in the case of planar-faced polyhedra, the well known result that a vertex or face connectivity graph represents a polyhedron unambiguously when the graph is triconnected and planar. The second result shows that when each face of a polyhedron of arbitrary genus has at most four edges, the polyhedron can be reconstructed uniquely. This extends the previous result that a polyhedron can be uniquely reconstructed when each face of the polyhedron is triangular. As a consequence of this result, faces are a more powerful representation than vertices for polyhedra whose faces have three or four edges. A result of the reconstruction algorithm is that high level features of the polyhedron are naturally extracted. Both results explicitly use the fact that the faces of the polyhedron are planar. It is conjectured that the spherical dual representation is unambiguous for polyhedra of any genus.

Keywords. Polyhedra; geometric dual; solid modeling; computer vision; CAD.

1. Introduction

A common means of representing a 3-dimensional object is through the abstraction known as a polyhedron. A *polyhedral surface* is a closed surface (a 2-manifold) that partitions Euclidean 3-space E^3 into 3 sets: (i) points inside the surface, (ii) points on the surface, and (iii) points outside the surface. A *polyhedron* is the set of points either inside or on a polyhedral surface. The *boundary* of the polyhedron is the polyhedral surface. Viewed combinatorially, the surface consists of a finite number of faces, edges, and vertices. Each edge is

shared by exactly two faces, and each edge has exactly two vertices as its endpoints. Each face is a connected open set. The edges incident to any vertex appear on the surface in a cyclic order around the vertex. Alternately, the faces incident to the vertex appear in a cyclic order around the vertex, and two faces adjacent in the order share an edge incident to the vertex. We only consider polyhedra with planar faces; that is, each face is contained in a plane.

Constructive solid geometry views a polyhedron in terms of point sets, while boundary representations, as the name indicates, characterize a polyhedron based on its boundary [19]. Since the boundary consists of entities of various dimensions—faces, edges, and vertices—there are various schemes for representing a polyhedron. Obtaining a polyhedron from its representation is termed *reconstruction.* If there is always a single polyhedron that can be obtained from any representation in the representation scheme, the polyhedron is *uniquely reconstructible* in that representation scheme. The problem we consider in this paper is the unique reconstruction of polyhedra in a representation scheme that encodes a minimal amount of information sufficient for reconstruction and that is useful for computer vision and other applications.

The spherical dual representation [20] is a representation scheme for polyhedra useful in both solid modeling and computer vision. The spherical dual representation of a polyhedron is a graph in which each face of the polyhedron is a node and is labeled by the equation of the plane containing the face. A node is connected by an arc to another if the two faces share an edge in the polyhedron. No ordering of the arcs around each node is specified. In fact, no explicit order information whatsoever is maintained. The spherical dual representation scheme can be viewed as the dual of the wire frame representation of polyhedra. Roach, Wright, and Ramesh [20] raise, but do not answer, the question of unique reconstructibility for this representation scheme.

In this paper, we investigate the reconstruction of a polyhedron from its spherical dual representation. It is well known that the wire frame representation (i.e., the vertex connectivity graph) of a polyhedron is ambiguous [16, 19]. Also, given either the wire frame or face connectivity graph of a genus 0 polyhedron, algorithms are known to *uniquely* reconstruct it only when the graph is triconnected [6]. We extend these algorithms to uniquely reconstruct any genus 0 polyhedron given its face connectivity graph (spherical dual representation). We prove that any spherical dual representation of degree at most 4 represents a polyhedron unambiguously. As a corollary to this result, we show that the face connectivity graph is not exactly the dual of the wire frame with regard to ambiguity. The results have an added importance since the spherical dual representation also has some interesting applications in computer vision. For example, the spherical dual representation provides some useful relationships between the representation of an object and its image under perspective projection [18].

The structure of the paper is as follows. The next section contains the necessary

graph theoretic and topological definitions. Section 3 reviews previous work in solid modeling representation and reconstruction. In Section 4, we develop our algorithm for uniquely reconstructing a genus 0 polyhedron given its spherical dual representation. Section 5 proves that the spherical dual representation of any polyhedron having maximum degree 4 is unambiguous. As a consequence, we have the theorem that for this restricted class of objects, the spherical dual representation is more powerful than the wire frame representation. The last section concludes with observations and conjectures.

2. Terms and definitions

A *graph* $G = (N, A)$ consists of a set of *nodes* N and a set of *arcs* A; each arc is an unordered pair of distinct elements from N. (We have chosen this non-standard terminology for undirected graphs—nodes and arcs instead of vertices and edges—to avoid confusion between the vertices and edges of a polyhedron and the nodes and arcs of the associated spherical dual representation. Nodes and arcs are generally used in the context of directed graphs.) If A is a multiset, that is, if an arc may occur several times, then G is a *multigraph*. Multiple arcs between the same pair of nodes are called *parallel arcs*.

A *path* P between nodes v_0 and v_k in a graph G is a sequence of nodes $v_0, v_1, \ldots, v_k$ such that $\{v_{i-1}, v_i\} \in A$, $1 \leq i \leq k$. Path P is a *simple path* if $v_0, v_1, \ldots, v_k$ are distinct. A *cycle* C in G is a path $v_0, v_1, \ldots, v_k$ such that $v_0 = v_k$. Cycle C is a *simple cycle* if $v_0, v_1, \ldots, v_{k-1}$ are distinct. A graph $G = (N, A)$ is *connected* if there exists a path between every pair of nodes in N. The number of arcs incident on a node v_i is called the *degree* of the node. Two arcs are said to be in *series* if they have exactly one node in common and if this node is of degree two. A node $v \in N$ is an *articulation point* of a connected graph $G = (N, A)$ if the subgraph induced by $N - \{v\}$ is not connected. A connected graph G is *biconnected* if G contains no articulation point. A *biconnected component* of G is a maximal induced subgraph of G which is biconnected. Let v_1, v_2 be a pair of nodes of a biconnected graph $G = (N, A)$; $\{v_1, v_2\}$ is a *separation pair* for G if the induced subgraph on $N - \{v_1, v_2\}$ is not connected. A biconnected graph G is *triconnected* if G contains no separation pair. A *triconnected component* of G is a maximal induced subgraph of G which is triconnected. Hopcroft and Tarjan [9] give an algorithm to find the triconnected components of a graph in time linear in the size of the graph.

The *genus* of an orientable, compact surface is the maximum number of non-intersecting simple closed curves that can be removed from its surface without disconnecting it. Thus the genus of a sphere is 0, and the genus of a torus is 1. In general, an orientable surface with g holes has genus g. The *genus* of a polyhedron is the genus of its surface.

A graph G is said to be topologically *embedded* in a surface S when it is drawn

on S such that no two arcs intersect except at their common nodes (see Gross and Tucker [4]). If a graph is embedded in a surface, the complement of its image is a finite set of connected components. A *region* of a topological embedding of G is a connected component of the complement of the image of G. The *genus* of a graph G is the genus of the orientable surface S of least genus such that G can be topologically embedded in S. A graph G is *planar* if G has an embedding in a plane (or, equivalently, in a sphere).

The *boundary* of a region f is the set of arcs in the (topological) closure of f. Two embeddings of a graph are *equivalent* when the boundary of a region in one embedding always corresponds to the boundary of a region in the other. The embedding of a graph on a surface is said to be *unique* if all its embeddings in that surface are equivalent. A planar graph has a unique planar embedding if and only if it is triconnected [26].

Given a connected graph G, a closed surface S, and an embedding $i: G \rightarrow S$, a *dual graph* G^* and a *dual embedding* $i^*: G^* \rightarrow S$ are defined as follows. For each region f of the embedding $i: G \rightarrow S$, place a node f^* in its interior. Then, for each arc e of the graph G, draw an arc e^* between the nodes just placed in the interiors of the regions containing e. The resulting graph with nodes f^* and arcs e^* is called the *dual graph* G^* for the embedding $i: G \rightarrow S$. The resulting embedding of the graph G^* in the surface S is called the *dual embedding*. Whitney [26] shows that a triconnected planar graph has a unique planar embedding and hence a *unique* dual.

3. Representations

In this section, we review some representations that have been used in geometric modeling and in computer vision, including the spherical dual representation. We also review known techniques for reconstructing a polyhedron from its representations.

3.1. Representations in geometric modeling

Geometric modeling is the art of creating data structures and algorithms capable of representing and calculating the three-dimensional physical shape of an object (Mantyla [16]). Requicha [19] identifies some important characteristics of a representation scheme for geometric modeling that have theoretical and practical implications:

(1) *Domain:* The domain of a representation scheme characterizes the descriptive power of the scheme; the domain is the set of entities representable in the scheme.

(2) *Validity:* The range of a representation scheme is the set of representations which are *valid*, that is, represent an actual 'solid'. A representation scheme is said to be *valid* if every representation in the scheme is valid.

(3) *Completeness:* A representation is *unambiguous* or *uniquely reconstructible* if it corresponds to a single object. A representation scheme is *complete* if all of its valid representations are unambiguous.

(4) *Uniqueness:* A representation of a solid is *unique* if it is the *only* representation for the solid in the scheme. A representation scheme is *unique* if all its valid representations are unique.

In this paper, we concentrate on the issue of whether a representation scheme is complete. As we are attempting to give a representation scheme using minimal amount of information that yields an unambiguous representation for each object, completeness is the central characteristic considered and the most difficult to prove.

The following are some of the common solid modeling schemes [16, 19, 22].

3.1.1. Wire frame representation

A wire frame model represents a solid object by representing its vertices and edges only. Each edge is typically represented by a six-tuple

$$\langle x_1, y_1, z_1, x_2, y_2, z_2 \rangle$$

giving the coordinates of the two endpoints (x_1, y_1, z_1) and (x_2, y_2, z_2) of the edge.

The main drawback of this representation scheme is its ambiguity. A wire frame model in general does not have enough information to represent an object uniquely (see Section 5). Characterizing it another way, two or more different objects can have the same set of edges. Thus wire frame representation is not a complete representation scheme.

3.1.2. *Constructive solid geometry*

A general form of the constructive solid geometry (CSG) approach is the *half-space* model. In this model, solids are represented by a finite number of simple point sets called half-spaces that are combined by the standard set operations of *union, intersection* and *difference.*

The CSG representation scheme is complete but not unique.

3.1.3. *Boundary representations*

Boundary representations represent a solid object by storing a description of its boundary. The boundary of an object divides space into two parts, one having finite volume and the other having infinite volume. If we assume that all objects have finite volume, then an object can be represented unambiguously by its boundary. The boundary is divided into a three-level hierarchy of entities: *faces, edges,* and *vertices.*

A widely used boundary representation is the solid modeling scheme based on Euler operators. Euler's formula for a convex polyhedron gives a relationship among the number of faces f, the number of edges e, and the number of vertices

v of a convex polyhedron: $v - e + f = 2$. Define a *loop* of a face to be a simple cycle of vertices and edges in the polyhedron that forms a connected component (in the topological sense) of the boundary of the face. The boundary of every face is composed of one external loop and zero or more internal loops. Euler's formula is generalized to an arbitrary polyhedron by introducing three additional parameters:

(1) The total number r of internal loops in the faces of the solid,

(2) The genus g of the solid, and

(3) The number s of disconnected components in a solid with a disconnected surface.

The general Euler's formula is $v - e + f = 2 * (s - g) + r$.

The operations used to construct the representation are called Euler operators because every operator used satisfies Euler's formula (e.g., two Euler operators are *mev*, for *make edge, vertex,* and *kef*, for *kill edge, face*). Mantyla [15] proves that Euler operators are valid and complete; that is, Euler operators create only meaningful models and every meaningful model can be constructed by Euler operators. Similar to the CSG scheme, boundary representations based on Euler operators are complete but not unique.

Representation schemes which are *both* unambiguous and unique are highly desirable because they are one-to-one mappings from the object space to the representation space. This implies that distinct representations in such schemes correspond to distinct objects, and therefore object equality may be determined by algorithms which compare object representations 'syntactically' [19]. Both the CSG scheme and the boundary representation scheme based on Euler operators are nonunique. However, unique boundary representations also exist.

3.2. Representations in computer vision

Object representations in computer vision are generally surface based. We review some of the representation schemes used in computer vision and introduce the spherical dual representation.

A *Gaussian map* is a function that maps the surface onto a unit sphere. Each point x on a surface is mapped to a point y on the unit sphere such that the surface normal at x equals the surface normal at y. The unit sphere in this context is called the *Gaussian sphere.* The image of a surface S under the Gaussian map is called the *Gaussian image* of S [10]. In case of a convex surface with positive curvature everywhere, no two points on the surface have the same normal but the surface is not recoverable from its Gaussian image since all distance and size information is lost in the encoding. In case of a general polyhedron, all points on a face map to the same point on the Gaussian sphere. The Gaussian image represents the orientation of the object only. Size and shape information is lost, making it impossible to reconstruct the object from the Gaussian image. A popular extension of the Gaussian image representation is the *extended Gaussian*

image. In this representation, each normal vector is weighted by the surface area of the corresponding face.

Other important surface representations exploit a duality between points and planes in three dimensions. Duality is an important concept in geometry [1, 8]. Dual space was originally proposed by Huffman as an aid in analyzing pictures of impossible objects [11] and later applied to interpreting general line drawings of polyhedral scenes [14, 12]. Huffman's version of duality involves associating the plane

$$ax + by + cz + d = 0 \tag{1}$$

with the point $(-a, -b, -d)$ in dual space. In addition to the duality between points and planes, there is also an induced duality between lines in (x, y, z)-space and lines in dual space. The dual of the line formed by the intersection of two planes is the line passing through the two points that are the duals of the planes. In Huffman's duality, only the first two coordinates of the dual point of a plane are related to orientation.

Gradient space, another duality representation, is formed by orthogonally projecting the dual points (a, b, d) onto the plane $d = 1$. Hence gradient space gives a two-dimensional or image representation, and the duality is between points and lines. Shafer [23] provides extensive analysis describing the advantages and uses of duality and gradient space in analyzing images for computer vision. Unfortunately, the interesting relationships between lines and points in the image and the lines and points in dual space are achieved under the assumption that the images are produced by orthogonal projection.

The *spherical dual representation* (SDR) dualizes planes into points by normalizing the constant d to -1 in Equation 1. This form of the dual transform is well known to mathematicians. Grunbaum [5] uses this transform to define a dual polyhedron when the given polyhedron is convex. Thus the plane

$$ax + by + cz - d = 0 \quad d \neq 0$$

is mapped to the dual point $(a/d, b/d, c/d)$ in spherical dual space [20]. We name this dual the spherical dual since this normalization has spherical symmetry about the origin as opposed to the cylindrical symmetry of Huffman's duality. To represent a polyhedron, each face is taken to be the point dual to the plane containing the face. The dual point is the node of a graph called the SDR of the polyhedron. The node f corresponding to face F is connected via an arc to the node f' corresponding to face F' if faces F and F' share an edge. Two faces sharing an edge cannot lie in the same plane. It is possible for two faces to share more than a single edge. The spherical dual representation does not explicitly represent such multiple adjacency and hence is not a multigraph. To accommodate multiple faces in the same plane, the spherical dual representation represents each face as a different node in the graph; that is, two nodes carry the same label (planar equation) if the corresponding faces lie in the same plane. Henceforth, we

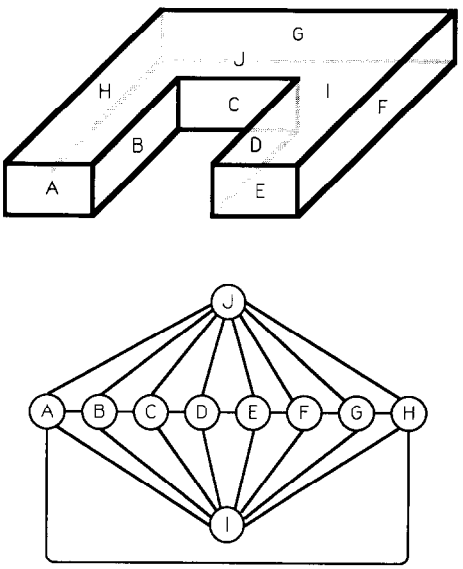

Fig. 1. An object and its SDR.

identify each node in the SDR with its associated face so that we can speak of a face as *being* a node of the SDR. Fig. 1 shows an object and its SDR (minus the planar equations). In view of the *graph* nature of the SDR, graph theoretic terms and operations apply to SDR. In fact, the spherical dual representation is the face connectivity graph of the polyhedron, where each node has an attached planar equation. In contrast to the wire frame, the SDR is always a connected graph as long as the surface of the polyhedron is connected.

Features of an object are high level abstractions that humans generally identify and operate with. Some examples of object features in manufacture, design, and recognition are boss, rib, blind hole, and through hole. Feature extraction at this abstract level is thus important in object recognition and geometric modeling systems. These features are further abstracted into *projecting features* and *depressions*. Falcidieno and Giannini [3] present a method for the automatic recognition and representation of shape-based features in a geometric modeling system. Loops in a face are the primary elements of this approach. The algorithm however requires the specification of the object as a face adjacency hypergraph, i.e., in addition to representing each face by a node and edges between faces by an arc between corresponding nodes, a hyperarc is defined for each vertex. Also each loop on a given face is organized into an ordered sequence of arcs. This also requires that their representation be a multigraph. The algorithm we present also extracts the shape features in terms of projecting features and depressions. Our method has the advantage that the specification of the object is only as a face adjacency graph rather than a face adjacency hypergraph. The loop information is extracted automatically. However, the domain of our algorithm is currently

restricted to genus 0 objects only. The ability to extract high level features automatically makes our representation very powerful in computer aided manufacture and computer vision applications.

3.3. Reconstruction techniques

Hanrahan [6] gives a linear time algorithm for the unique reconstruction of a genus 0 polyhedron given its wire frame representation. This algorithm, however, requires the wire frame input of the polyhedron to be triconnected and planar in the graph theoretic sense. The faces of the polyhedron correspond to the regions in the unique planar embedding of the wire frame.

Markowsky and Wesley [17] present an algorithm that generates all polyhedra with a given wire frame. This explicitly uses topological and geometric information by forcing the final faces to be planar. Human intervention is required to choose one of the several polyhedra reconstructed from such a wire frame representation.

Weiler [25] enumerates those boundary representations of polyhedra that are sufficient for unique reconstruction. Making use of Edmonds' Theorem [2], Weiler shows that knowing the *ordered* set of edges around each vertex, or each edge, or each face of a polyhedron is sufficient information for the reconstruction of any polyhedron. Weiler also states that a representation without order has insufficient information for unique reconstruction. Since Weiler's argument is based on the number of different embeddings of a graph on a surface, the same conclusion is reached if the polyhedra are restricted to genus 0. However, Weiler's argument applies also to more general polyhedra, i.e., solid objects having non-planar faces as well. Markowsky and Wesley [17] is the only reference that deals with reconstruction of planar faced objects from unordered boundary representations.

Later, we show unique reconstructibility for genus 0 polyhedra when represented by SDR, a representation without order information. We also prove that a polyhedron of any genus that only has faces with at most 4 edges is uniquely reconstructible from its *SDR*.

A different approach to reconstruction of convex polyhedra is suggested by Minkowski's Theorem [5]. Minkowski uniquely characterizes, up to a translation, any convex polyhedron by the *area* of its faces and their orientations. Using the Minkowski and Brunn–Minkowski Theorems [5], Little [13] solves the problem of reconstructing the polyhedron, given its extended Gaussian image, by solving a constrained minimization problem. The domain of Little's algorithm is the same as that of Minkowski's Theorem; it fails to reconstruct non-convex polyhedra.

The most important result on the realization of a polyhedron from its wire frame is Steinitz's Theorem [5]. The theorem states that *a graph G is realizable as a convex polyhedron if and only if G is planar and triconnected.* In the case of reconstruction of polyhedra, this theorem can be used for all combinatorially

convex polyhedra. A polyhedron is *combinatorially convex* if its wire frame is planar and triconnected and the polyhedron has genus 0. From Whitney's Theorem [26], every combinatorially convex polyhedron is uniquely reconstructible from its face connectivity graph, i.e., its SDR. In the next section, we extend unique reconstructibility to *every* genus 0 polyhedron.

4. Reconstruction of genus 0 polyhedra

In this section, we present an algorithm RECONSTRUCT that uniquely reconstructs any genus 0 polyhedron P from its spherical dual representation. RECONSTRUCT first builds a graph for each face in P and then extracts the vertices of each boundary of the face from that graph.

Let $\mathrm{SDR} = (N, A)$ be the spherical dual representation of the genus 0 polyhedron P. Let $\mathcal{P}(f)$ be the plane containing the face $f \in N$. Each face f of P consists of a bounded, connected region in $\mathcal{P}(f)$ that has one or more cycles of edges and vertices of P as boundary. If f is bounded by t cycles, then f has exactly $t-1$ holes. To reconstruct P, it suffices to determine all the bounding cycles of all faces. Let $\mathcal{F}(f)$ be the set of faces that are adjacent to f in *SDR*. If $f^* \in \mathcal{F}(f)$, then $\mathcal{L}(f, f^*) = \mathcal{P}(f) \cap \mathcal{P}(f^*)$ is an infinite line within $\mathcal{P}(f)$ that contains the (one or more) edges of P that are shared by f and f^*.

Suppose that $f^*, f^{**} \in \mathcal{F}(f)$ have the property that lines $\mathcal{L}(f, f^*)$ and $\mathcal{L}(f, f^{**})$ are not parallel. Then $\mathcal{L}(f, f^*)$ and $\mathcal{L}(f, f^{**})$ intersect at a point $\mathcal{T}(f, f^*, f^{**})$ within $\mathcal{P}(f)$. Let $\mathcal{Q}(f)$ be the set of all such intersections within $\mathcal{P}(f)$. Then every vertex v of P that is incident to f is an element of $\mathcal{Q}(f)$. In general, $\mathcal{Q}(f)$ contains many points that are not vertices of P. A necessary condition for a point $p \in \mathcal{Q}(f)$ to be a vertex of P is that there exist a *defining cycle* $f, f_1, \ldots, f_k$ in SDR such that $p \in \mathcal{Q}(f_i)$, $i = 1, \ldots, k$. (Generalizing the $\mathcal{T}$ notation, we write $p = \mathcal{T}(f, f_1, \ldots, f_k)$.) If p is indeed a vertex of P, then there is, of course, a *simple* defining cycle for p which is the cycle of faces incident to p. However, a given point p may have many defining cycles. That the existence of a defining cycle is not sufficient for a point p to be a vertex is shown by the example in Fig. 2. This example is a truncated tetrahedron, where the face E has cut off the top vertex p of the original tetrahedron. The point p is not a vertex of the truncated tetrahedron, yet $p \in \mathcal{Q}(A)$, $p \in \mathcal{Q}(B)$, $p \in \mathcal{Q}(C)$, and A, B, C is a cycle in SDR.

If a point $p \in \mathcal{Q}(f)$ meets the above necessary condition (of having a defining cycle), call p a *near-vertex of* f. The minimal subgraph $\mathcal{NF}(f)$ of SDR that contains all defining cycles of every near-vertex of f is the *near-face graph* of f. Clearly, every node in $\mathcal{F}(f)$ is also adjacent to f in $\mathcal{NF}(f)$.

An outline of an algorithm for constructing $\mathcal{NF}$(f) for all $f \in N$ follows. Let $\mathcal{J}$ be the set of all lines in 3-dimensional space defined by edges in P:

$$\mathcal{J} = \{\mathcal{L}(f_1, f_2) \mid (f_1, f_2) \in A\}.$$

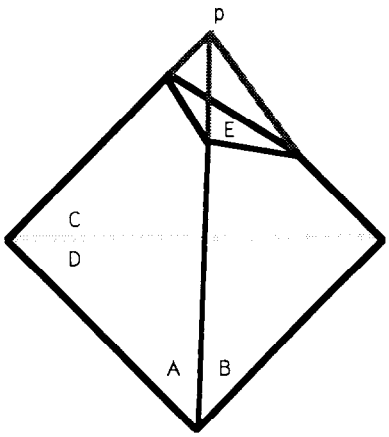

Fig. 2. Truncated tetrahedron.

Let $\mathcal{J}$ be the set of all pairwise intersections of two distinct lines in $\mathcal{J}$ such that each intersection is a single point:

$$\mathcal{J} = \{\mathcal{T}(f_1, f_2, f_3, f_4) \mid \mathcal{L}(f_1, f_2), \mathcal{L}(f_3, f_4) \in \mathcal{J} \text{ and } |\mathcal{L}(f_1, f_2) \cap \mathcal{L}(f_3, f_4)| = 1\}.$$

Clearly, for each $f \in N$, we have $\mathcal{Q}(f) \subset \mathcal{J}$. The calculation of $\mathcal{J}$ requires $O(|A|^2) = O(|N|^2)$ time. For each $p \in \mathcal{J}$, build a subgraph $\mathcal{H}(p)$ of SDR induced by this set of arcs:

$$\{(f_1, f_2) \mid p \in \mathcal{L}(f_1, f_2)\}.$$

In $O(|N|)$ time per $p \in \mathcal{J}$, eliminate from $\mathcal{H}(p)$ any arcs that do not appear in a cycle of $\mathcal{H}(p)$ and any isolated nodes; the resulting graph is $\mathcal{H}'(p)$. Calculate the biconnected components of each $\mathcal{H}'(p)$. For a particular $f \in N$, consider every $\mathcal{H}'(p)$ that contains f; every biconnected component of $\mathcal{H}'(p)$ that contains f is a subgraph of $\mathcal{NF}(f)$. In fact, all of $\mathcal{NF}(f)$ is obtained by taking the union of all such biconnected components from every $\mathcal{H}'(p)$ that contains f. The calculation of $\mathcal{NF}(f)$ for all $f \in N$ is accomplished in $O(|N|^3)$ time. We emphasize that this is a worst case time complexity; under reasonable assumptions on the sizes of each $\mathcal{NF}(f)$ and each $\mathcal{H}(p)$, the time complexity can be reduced to $O(|N|^2)$.

Shortly, we will be embedding subgraphs of $\mathcal{NF}(f)$ in the plane and reading off the vertices incident to f from the regions of the embeddings. Any node of degree two in $\mathcal{NF}(f)$ has no effect on these embeddings and can be eliminated by *series reduction* (replace the node and its two incident arcs by single arc; see Gross and Tucker [4]). If any parallel arcs are introduced by series reduction, all but one can be eliminated by *parallel reduction.*

There is one last kind of reduction that can be applied to $\mathcal{NF}(f)$ without losing the ability to recover the vertices on the boundary of f. Suppose $\{f_1, f_2\} \subset N - \{f\}$ is a separation pair of $\mathcal{NF}(f)$. Let C be any component of $\mathcal{NF}(f) - \{f_1, f_2\}$ that does not contain f. Let f^* be a node in C. Since every defining cycle containing f^* must pass through f, f_1, and f_2, the plane f^* passes through

$\mathcal{T}(f, f_1, f_2)$. Therefore, *every* node of C passes through $\mathcal{T}(f, f_1, f_2)$; in a geometric sense, the nodes of C give only redundant information. A *separation-pair reduction* deletes C and adds an arc between f_1 and f_2. By the above discussion, such a reduction does not affect the information available for recovering vertices on the boundary of f.

The *face graph* SDR(f) of f is $\mathcal{NF}(f)$ that has been reduced as much as possible by series, parallel, and separation-pair reductions. As SDR is planar, $\mathcal{NF}(f)$ is a subgraph of SDR, and SDR(f) is a reduction of $\mathcal{NF}(f)$, SDR(f) is also planar. Also, every node in $\mathcal{F}(f)$ is a node of SDR(f). As calculating all the triconnected components of a graph can be accomplished in linear time [9], the reduction of $\mathcal{NF}(f)$ to SDR(f) can be accomplished in $O(|N|)$ time. While finding the triconnected components, the algorithm in [9] also finds all separation pairs.

Since P has genus 0, SDR(f) gives us all the information necessary to determine the bounding cycles of f. For example, the number of bounding cycles is just the number of biconnected components of SDR(f). This is illustrated in Fig. 3 where face F has two bounding cycles, and SDR(F) has two biconnected components. Observe also that F is the sole articulation point of SDR(F). This observation is formalized in the following theorem.

Theorem 1. *Let* SDR $= (N, A)$ *be the spherical dual representation of a genus* 0 *polyhedron* P, *and let* f *be a node of* SDR. *If* SDR(f) *contains an articulation*

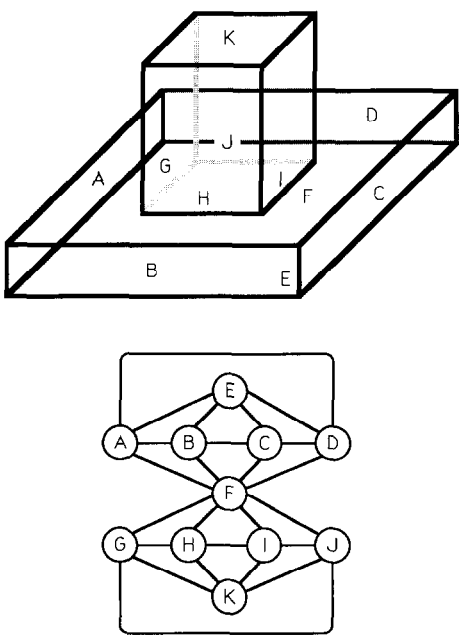

Fig. 3. A polyhedron with an articulation point in its SDR.

point, then f is the only articulation point. The number of bounding cycles of f equals the number of connected components of $\mathrm{SDR}(f)-f$.

Proof. By the definition of $\mathrm{SDR}(f)$, every node f^* in $\mathcal{NF}(f)-\{f\}$ has two vertex disjoint paths to f in $\mathrm{SDR}(f)$. Therefore, only f can be an articulation point of $\mathrm{SDR}(f)$.

Assume that f has t bounding cycles. Define an equivalence relation $\equiv$ on $N-\{f\}$ such that $f^* \equiv f^{**}$ if there exists a curve on (the surface of) P that goes from a point in the interior of f^* to a point in the interior of f^{**} without passing through the closure of f (that is, the curve avoids f and its boundary). Because P has genus 0, $\equiv$ has exactly t equivalence classes. $\mathcal{NF}(f)-f$ has one component for each equivalence class. It is easy to see that series, parallel, and separator-pair reductions apply independently to each component of $\mathcal{NF}(f)-f$. Thus, $\mathrm{SDR}(f)-f$ has the same number of components as $\mathcal{NF}(f)-f$, namely t. □

As SDR is planar but not necessarily triconnected, SDR does not have, in general, a unique embedding in the plane, whose dual would be the wire frame of P. A first approach that is doomed to failure is to decompose SDR into its triconnected components, embed each in the plane, and somehow read off the structure of P from these embeddings. The failure of this approach is illustrated by the polyhedron in Fig. 4, shown with its SDR. The triconnected components of SDR are shown embedded in the plane in Fig. 5. There is no region in any of the embeddings that corresponds to the vertex of P shared by the faces F, C, I,

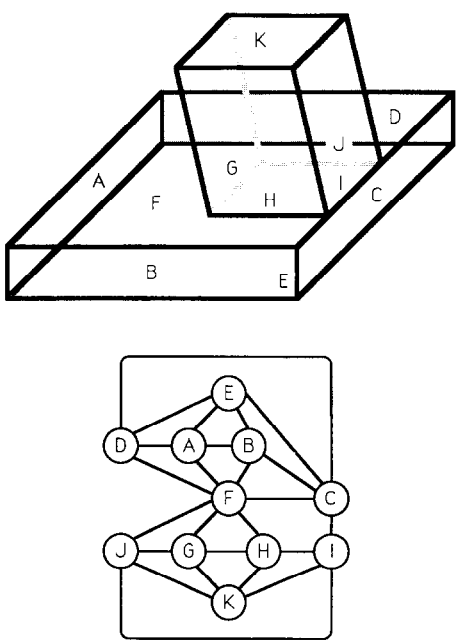

Fig. 4. A polyhedron whose SDR is not triconnected.

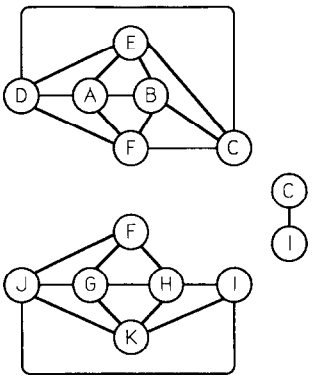

Fig. 5. The corresponding triconnected components.

and H, nor to the vertex shared by the faces F, C, I, and J. However, there is a 'false' vertex indicated by the region bounded by the cycle of faces F, H, I, J.

In view of this failure, we turn to face graphs for a solution. From the proof of Theorem 1, we know that if SDR(f) has biconnected components $C_1, C_2, \ldots, C_t$, then each C_i contains f and corresponds precisely to one of the bounding cycles of f. As each C_i may be processed separately to determine its corresponding bounding cycle, we henceforth assume that SDR(f) contains only one biconnected component, namely SDR(f) itself. SDR(f) can be decomposed into its biconnected components in linear time using depth-first search (Tarjan [24]).

The bounding cycle of f is given by the sequence of vertices defining it, say,

$$v_1, v_2, \ldots, v_k.$$

Each v_i has a defining cycle given by the actual faces that are incident to v_i:

$$f, f_{i,1}, f_{i,2}, \ldots, f_{i,r(i)},$$

where $f_{i,r(i)} = f_{i+1,1} \in \mathcal{F}(f)$, for $1 \leq i \leq k-1$, and $f_{k,r(k)} = f_{1,1} \in \mathcal{F}(f)$. From these cycles we derive another cycle in $\mathcal{NF}(f)$ that avoids f but otherwise goes along the bounding cycle of f:

$$f_{1,1}, f_{1,2}, \ldots, f_{1,r(1)} = f_{2,1}, \ldots, f_{2,r(2)} = f_{3,1}, \ldots,$$
$$f_{i-1,r(i-1)} = f_{i,1}, f_{i,2}, \ldots, f_{i,r(i)} = f_{i+1,1}, \ldots, f_{k,r(k)} = f_{1,1}.$$

This cycle in $\mathcal{NF}(f)$ is reduced to a cycle $\mathcal{C}(f)$ in SDR(f), called the *neighborhood cycle of* f. Note that $\mathcal{C}(f)$ need not be a simple cycle.

It is now our task to determine the vertices that occur on the bounding cycle of f and the order in which they occur. If SDR(f) is triconnected, then it has a unique planar embedding. Suppose that the order of the nodes in $\mathcal{F}(f)$ about f in this embedding is $f_1, f_2, \ldots, f_k$ (note that these are not necessarily distinct). These correspond to the vertices

$$v_1 = \mathcal{T}(f, f_1, f_2),\ v_2 = \mathcal{T}(f, f_2, f_3), \ldots, v_k = \mathcal{T}(f, f_k, f_1),$$

in that order, defining the bounding cycle of f. Call this cycle of vertices the *cycle induced by the embedding*.

If SDR(f) is not triconnected, then it may not be true that all embeddings induce the same cycle of vertices or even that there exists an embedding that induces a cycle equal to the bounding cycle of f. We observe the following.

Lemma 2. *If* SDR(f) *is biconnected, then any separation pair of* SDR(f) *contains* f.

Proof. Since there is no separation-pair reduction that can be applied to SDR(f), the Lemma follows. □

Thus any separation pair of SDR(f) has the form (f, f^*), where f^* is a node of SDR(f) which may or may not be adjacent to f. Call f^* a *separation partner* of f. All separation partners of f can be identified by finding the articulation points of SDR(f) $-f$ in linear time using depth-first search. For example, Fig. 6 shows the face graph of the face F in the polyhedron of Fig. 4. The separation partners of f are faces C and I. Note that F and C are adjacent in SDR(f), while F and I are not.

The example in Fig. 4 is a degenerate one in that $\mathcal{P}(F)$, $\mathcal{P}(C)$, and $\mathcal{P}(I)$ intersect in a single line. Fig. 7 shows another example in which there is a single separation pair whose nodes are not adjacent. The polyhedron is a box with a raised pyramid (faces C, D, E, and F) on its front face. Faces A and B are distinct faces that reside in the same plane and that are not adjacent in SDR. Face B is a separation partner of A (and vice versa). Note that A and B share (are incident to) both vertices v_1 and v_2. This is a general phenomenon:

Lemma 3. *If f^* is a separation partner of f and $f^* \notin \mathcal{F}(f)$, then f and f^* have two or more shared vertices but no shared edges.*

Proof. Since $f^* \notin \mathcal{F}(f)$, f and f^* do not share an edge.

Since f^* is in at least one cycle of SDR(f) that also contains f, and f^* is an articulation point of SDR(f) $-f$, there is some region containing both f and f^* in every planar embedding of SDR(f). Therefore, f and f^* share at least one vertex. In particular, f^* occurs in the neighborhood cycle $\mathcal{C}(f)$. Since every node in

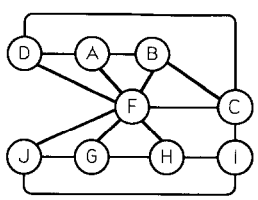

Fig. 6. The face graph SDR(F).

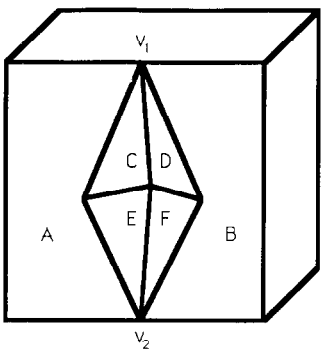

Fig. 7. A separation pair whose nodes are not adjacent.

$\mathcal{F}(f)$ occurs in $\mathcal{C}(f)$, and f^* is an articulation point of $\mathrm{SDR}(f)-f$, f^* must occur at least twice in $\mathcal{C}(f)$. Otherwise, the removal of f^* from $\mathrm{SDR}(f)-f$ leaves $\mathcal{C}(f)$, and hence $\mathrm{SDR}(f)-f$, connected.

Since f^* occurs at least twice in $\mathcal{C}(f)$, f and f^* must share two or more vertices. □

We can make a stronger observation. Let f^* be a separation partner of f. Suppose SDR* is a connected component of $\mathrm{SDR}(f)-f-f^*$. Then there exists a closed curve contained in the closure (in the topological sense) of $f \cup f^*$ that separates P into two regions, each homeomorphic to a disk, such that SDR* is contained wholly in one of the regions and the remainder of $\mathrm{SDR}(f)-f-f^*$ is contained wholly in the other region.

SDR* may or may not contain a separation partner of f. If it does not, then we can determine the boundary between f and SDR* as follows. Construct the *decomposition graph* $\mathrm{SDR}^*(f^*)$ by taking the subgraph of $\mathrm{SDR}(f)$ induced on f, f^*, and the nodes of SDR*, and add the arc (f, f^*) if $f^* \notin \mathcal{F}(f)$. The following observation is key.

Lemma 4. *If* SDR* *contains no separation partner of* f, *then* $\mathrm{SDR}^*(f^*)$ *is triconnected.*

Proof. Suppose $\mathrm{SDR}^*(f^*)$ is not triconnected and that (f_1, f_2) is a separation pair of $\mathrm{SDR}^*(f^*)$. Then (f_1, f_2) is also a separation pair of $\mathrm{SDR}(f)$. Since $\mathrm{SDR}(f)$ is separation-pair reduced, either f_1 or f_2 equals f and the other is a separation partner of f. □

Hence, $\mathrm{SDE}^*(f^*)$ has a unique planar embedding. The order of arcs in $\mathcal{F}(f)$ around f in this embedding exactly gives the order in which faces of $\mathrm{SDR}^*(f^*)$ appear in the boundary between f and SDR*. As is true when $\mathrm{SDR}(f)$ is triconnected, the cycle induced by the embedding gives the (cyclic) order of vertices and edges that form the bounding path. The cycle breaks into a path at the arc (f, f^*).

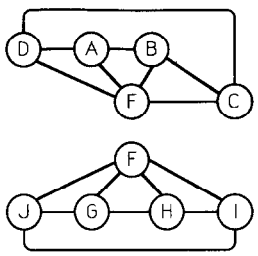

Fig. 8. The decomposition graphs of SDR(F).

For example, if we apply this decomposition to the face graph SDR(F) in Fig. 6, we obtain two decomposition graphs as shown in Fig. 8. The arc (F, I) has been added to the lower decomposition graph. From the two planar embeddings, we learn the ordering of the vertices and edges in each of the two intermediate cycles C_1 and C_2 on F. Intermediate cycle, C_1 has faces A, B, C, and D in that order, while intermediate cycle C_2 has faces G, H, I, and J in that order. Since (F, C) is a separation pair, the edge $\{\mathcal{T}(F, C, B), \mathcal{T}(F, C, D)\}$ that is dual to the arc (F, C) cannot be present as a whole in the boundary of F (in Fig. 4, we see that it breaks into two edges and a missing center along the boundary between F and C). Similarly, the edge $\{\mathcal{T}(F, I, H), \mathcal{T}(F, I, J)\}$ that is dual to (F, I) cannot be presented as a whole in the boundary of F (in Fig. 4, it is in fact the missing center). The symmetric difference of these edges forms the actual pair of edges between F and C. Hence, these two cycles are mated geometrically, obtaining a unique cycle $C = (C_1 \cup C_2 - C_1 \cap C_2)$, which is the bounding cycle of F. By *mating geometrically*, it is meant that the cycles are viewed as point sets in Euclidean space and the union and difference operations are performed over this point sets. Viewed in terms of paths, they begin and end at the separation partners of F.

For any SDR(f) that is not triconnected, there always exists an SDR* that contains no separation partner for f (examine the tree of biconnected components and articulation points of SDR(f) $-f$ to find a biconnected component that is a leaf). We can then determine the subpath of the bounding cycle of f that is between f and SDR* by the above decomposition method. Once the subpath is determined, we would like to remove SDR* from further consideration. This can be done by replacing SDR(f) by the *reduced graph* SDR(f) $-$ SDR* with the arc (f, f^*) added, if $f^* \notin \mathcal{F}(f)$. In geometric terms, this reduction amounts to removing the feature of P that corresponds to SDR* and replacing it by an edge in P shared by f and f^*. Geometrically this may not always work, as witnessed by the polyhedron in Fig. 7, where faces A and B are in the same plane and therefore cannot share an edge. However, combinatorially the reduction does work. As stated earlier, the reduction reveals the structure of the polyhedron at a high level.

The strategy for finding the bounding cycle of SDR(f) now is clear. Iteratively find a decomposition graph SDR*(f^*) that contains no separation partner of f,

determine the corresponding bounding path, and reduce SDR(f). Once SDR(f) is reduced to a triconnected graph, construct the bounding cycle of f by gluing the subpaths together at the separation partners of f. This completes the description of the processing of each biconnected component of SDR(f).

In summary, the algorithm RECONSTRUCT consists of the following steps, applied to each face f.

1. Form the set $\mathcal{Q}(f)$.
2. Determine $\mathcal{NF}(f)$ and reduce it to SDR(f).
3. Decompose SDR(f) into its biconnected components; say these components are $\text{SDR}_1, \text{SDR}_2, \ldots, \text{SDR}_k$, $k \geq 1$.
4. For each biconnected component SDR_i, determine the bounding cycle of f corresponding to SDR_i, using the decomposition graph strategy.

The time complexity is dominated by the determination of $\mathcal{NF}(f)$ for all $f \in N$; this step has time complexity $O(|N|^3)$, as discussed earlier in this section. Determining $\mathcal{NF}(f)$ and reducing it to SDR(f) only takes linear time. Finding the biconnected components is the same complexity as finding the articulation points of SDR(f) $-f$. Embedding a triconnected component is again a linear time operation. As the size of SDR(f) may be $\Theta(|N|)$, the time complexity of these steps for each face is $O(|N|)$. The total time complexity for RECONSTRUCT is $O(|N|^3)$, though we expect that it is typically much less.

Theorem 5. *Algorithm* RECONSTRUCT *uniquely reconstructs any genus* 0 *polyhedron given its SDR. The time complexity of* RECONSTRUCT *is* $O(|N|^3)$.

Algorithm RECONSTRUCT successfully reconstructs some, but not all, polyhedra of genus greater than zero. If P is a polyhedron of arbitrary genus, then it is possible that, for some face f, SDR(f) is not even planar. This occurs when (one or more) cycles in SDR(f) pass through (one or more) holes in P. Also, a face f with multiple bounding cycles need not even be an articulation point in SDR(f) if a hole of P passes through f.

RECONSTRUCT can be modified to successfully reconstruct more polyhedra of genus greater than zero as follows. Typically, the faces in some non-empty subset of N can be successfully reconstructed by the steps in RECONSTRUCT. Boundary information from these reconstructed faces can be shared with neighboring faces that are not immediately reconstructible, perhaps making them reconstructible in the process. If such information sharing propagates to all faces of P, then the modified RECONSTRUCT successfully reconstructs P. We have found that this modified algorithm is capable of reconstructing a number of 'hard' polyhedra that we had proposed as potential counterexamples to unique reconstructibility for polyhedra of higher genus.

The SDR shown in Fig. 9 is a graph that cannot be solved by the above (modified) algorithm. The genus of this graph is 1. The SDR(f) of each face of

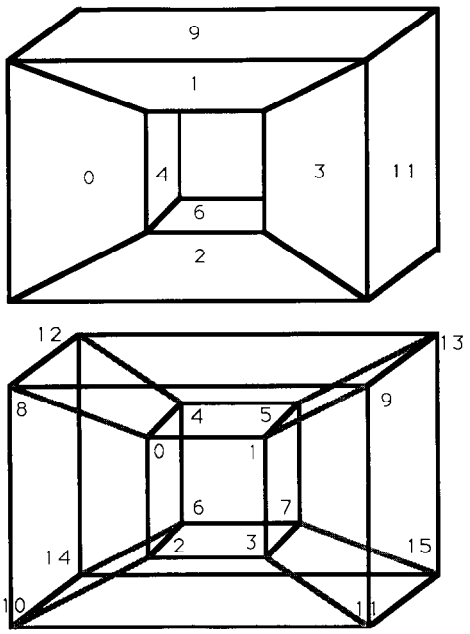

Fig. 9. A degree four polyhedron and its SDR.

the polyhedron is non-planar and triconnected (each SDR(f) is homeomorphic to K_5). Hence not even a single face can be reconstructed via planar embedding of its SDR(f). In the next section, however, we show that the polyhedron in Fig. 9 does have a unique reconstruction from its SDR. Incidentally, the vertex connectivity graph of the polyhedron in Fig. 9 represents three distinct polyhedra.

5. SDRs of maximum degree 4

In this section, we show that an SDR of degree four represents a unique polyhedron. We also provide an algorithm to reconstruct the polyhedron from its SDR.

Considers SDRs of degree 4. Since each face is connected to four other faces, each face must be a quadrilateral. If the bounding cycle of each face is determined, then the polyhedron represented by this SDR is reconstructed. We study the different quadrilaterals that can be formed by the adjacent faces. Consider the arrangement of four lines in a plane. In case of degeneracies among the four lines, there may not be any quadrilateral formed by these lines, for example when three of these lines are concurrent. On the other hand, if two of the lines are parallel, at most one quadrilateral is formed and the reconstruction of this face is unambiguous.

The four lines defined by the face adjacencies form two different quadrilaterals when the four lines are in general position. Refer to Fig. 10 which depicts the

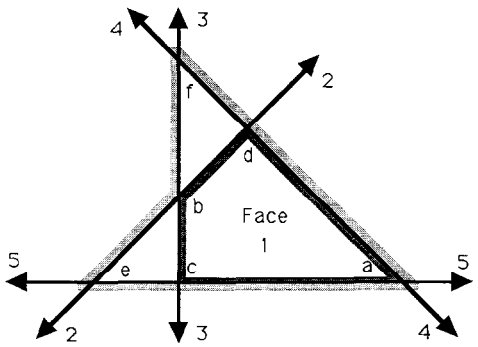

Fig. 10. Two realizations of face 1.

plane containing face 1 and the four lines formed by intersection with the planes containing faces 2, 3, 4, and 5. There are two possible interpretations for the boundary of each face. One interpretation has vertices *a, f, b,* and *e,* while the other interpretation has vertices *a, d, b,* and *c.* Two vertices, *a* and *b,* are present in both interpretations. Vertex *a,* called the *fixed vertex,* has its context unchanged, i.e., when we follow the boundary of the two polygons in the same direction, the line segments occur in the same order. Vertex *b,* called the *reflex vertex,* has its context reversed. The internal angle at the reflex vertex changes from being a convex angle ($\angle dbc$) in one interpretation to concave ($\angle fbe$) in the other. It is immediately clear from Fig. 10 that fixing any one of the remaining four vertices determines the polygon unambiguously. These four vertices are termed the *transient* vertices. Vertices on each of the lines through the fixed vertex and nearer the fixed vertex are called *intruded vertices* and those farther away are called *extruded vertices.* Thus in face 1 in Fig. 10, vertices *c* and *d* are intruded vertices and *e* and *f* are extruded vertices.

Given an arrangement of four lines in a plane, the following procedure can be used to classify a vertex as reflex, fixed, or transient. Four lines in general position determine six (potential) vertices. The vertex that lies inside the convex hull defined by the six vertices is the reflex vertex. It is easy to see that there is exactly one such vertex. In Fig. 10, the vertex formed by lines 2 and 3 is the reflex vertex. The vertex defined by the two lines not involved in defining the reflex vertex is the fixed vertex. Again, referring to Fig. 10 the fixed vertex is defined by lines 4 and 5. The other four vertices are transient vertices. Of the transient vertices, the two vertices forming vertices of the convex hull are extruded vertices and the other two are intruded vertices. It is clear that the classification of a vertex is a constant time operation. Note that the classification of a vertex is only with respect to a specific face; its classification may be different on a different face.

Observe that in any particular interpretation, the transient vertices are either both intruded or both extruded. Observe also that the internal angle at the reflex vertex is less than 180° (convex) if the intruded vertices are chosen and is greater than 180° (concave) if the extruded vertices are chosen.

Now we study the constraints provided by the adjacent faces. Observe that knowing any edge in the quadrilateral determines the complete quadrilateral since each edge has exactly one transient vertex incident to it. Consider a face f_j adjacent to a face f_i. If either the reflex vertex on face f_i or the fixed vertex on face f_i is a transient vertex on face f_i, then face f_j is completely determined. This in turn determines face f_i and the other faces adjacent to face f_i are completely determined. On the other hand, if the reflex vertex on face f_i is the fixed vertex on face f_j, face f_i is completely determined and consequently all the faces adjacent to face f_i.

Thus the only possibility for an ambiguous interpretation occurs when the fixed vertex on face f_i is also the fixed vertex on an adjacent face f_j and the reflex vertex on face f_i is the reflex vertex on an adjacent face f_k.

Suppose face f_j is adjacent to face f_i and they share the reflex vertex. If the convex internal angle at the reflex vertex in face f_i is consistent with the convex internal angle at the reflex vertex in face f_j, then again there is no ambiguity in determining the vertices and edges on faces f_i and f_j. This is evident when one observes that no two adjacent faces of a polyhedron can simultaneously have an internal angle greater than 180° at a common vertex.

Finally, ambiguity can continue to persist when the adjacent faces at a reflex vertex form an alternating sequence of convex and concave angles. Thus the transient vertices switch their classifications between adjacent faces, i.e., an intruded vertex on one face is an extruded vertex on the adjacent face.

Now we show that there cannot be ambiguity at a reflex vertex which has an odd number of faces incident on it. Since there is a switch from convex to concave angles between the two interpretations, ambiguity at a reflex vertex with an odd number of faces would imply that there are at least two adjacent faces each of which has an internal angle greater than 180°. This is impossible in polyhedra. Thus there cannot be any ambiguity at a reflex vertex where an odd number of faces come together.

The remaining case to be considered is when an even number of faces come together at a reflex vertex, and all the nodes are of degree 4. We now prove that for this case also there is only one interpretation. (A proof for a simpler case can be found in Heath, Paripati, and Roach [7].)

Lemma 6. *An* SDR *of degree* 4 *with an even number of faces forming a reflex vertex represents a unique polyhedron.*

Proof. Let f_i be a face involved in forming a reflex vertex and n the degree of the reflex vertex. Without loss of generality, we may assume that the reflex vertex is at the origin. The fixed vertex on f_i is p_i, $i = 1, \ldots, n$, while the intruded vertices are q_{2i-2} and q_{2i-1}. The extruded vertices are q_{2i} and q_{2i-3}. Subscripts are computed modulo $2n$ with the caveat that 0 is represented by $2n$. The subscript arithmetic is unaffected by this variation. As an illustration, if there are 6 faces

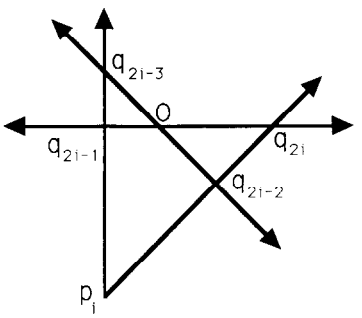

Fig. 11. Vertex labeling of the realizations of face f_i.

meeting at a reflex vertex, and we are considering the vertices on face 1, then the intruded vertices are q_{12} and q_1, and the extruded vertices are q_2 and q_{11} respectively. The line through the odd numbered vertices q_{2i-3} and q_{2i-1} passes through p_i and is formed by the intersection of f_i with one adjacent face. Similarly, the line through the even numbered vertices q_{2i-2} and q_{2i} passes through p_i. See Fig. 11. From the n faces meeting at the reflex vertex, the following relationships follow

$$t_{2i-1}(q_{2i-2} - p_i) = q_{2i} - p_i, \tag{2}$$

$$t_{2i}(q_{2i-1} - p_i) = q_{2i-3} - p_i, \tag{3}$$

$$u_i q_{2i-1} = q_{2i}, \tag{4}$$

where $t_{2i-1} > 1$, $t_{2i} > 1$, $u_i < 0$, $i = 1, \ldots, n$. Substitute q_{2i} for $i = 1, \ldots, n$, by $u_i q_{2i-1}$ from Equation 4 into Equations 2 and 3 obtaining

$$(t_{2i}(t_{2i-1} - 1) + (t_{2i} - 1)u_i)q_{2i-1} - ((t_{2i-1} - 1) + (t_{2i} - 1)t_{2i-1}u_{i-1})q_{2i-3} = 0. \tag{5}$$

From the geometry of the lines in Fig. 11 and the fact that the origin is the reflex vertex, the coefficient of each point q_i in Equation 5 must be zero, i.e.,

$$t_{2i}(t_{2i-1} - 1) + (t_{2i} - 1)u_i = 0, \tag{6}$$

$$(t_{2i-1} - 1) + (t_{2i} - 1)t_{2i-1}u_{i-1} = 0. \tag{7}$$

Replace u_i, $i = 1, \ldots, n$ in Equation 7 using Equation 6 obtaining

$$\frac{(t_{2i} - 1)}{(t_{2i-1} - 1)} = t_{2i}t_{2(i+1)-1}\frac{(t_{2(i+1)} - 1)}{(t_{2(i+1)-1} - 1)}$$

which yields

$$\frac{(t_2 - 1)}{(t_1 - 1)} = t_1 t_2 \cdots t_{2n}\frac{(t_2 - 1)}{(t_1 - 1)},$$

and finally,

$$t_1 t_2 t_3 \cdots t_{2n} = 1.$$

This is a contradiction, since $t_i > 1$, $i = 1, \ldots, 2n$. The lemma follows. □

Thus there cannot exist an even number of faces forming a reflex vertex such that every face has two interpretations. Since there is no ambiguity at the reflex vertex, all the faces of the polyhedron are determined unambiguously.

As pointed out earlier, there cannot be any ambiguity in the interpretation of a quadrilateral formed by four arbitrary lines if any one of the transient vertices is known. Thus if any degree 4 face f_i has a degree three face adjacent to it, f_i is unambiguous. All degree 4 faces adjacent to f_i can then be resolved. Hence the SDR of any polyhedron of maximum degree 4 with at least one node of degree 3 also represents a unique polyhedron and we have the following theorem.

Theorem 7. *The* SDR *of a polyhedron of any genus is unambiguous if its degree is at most* 4.

The previous discussion also provides an algorithm for reconstruction. Ambiguity at one reflex vertex is resolved using Equations 4–7. This can be done in time proportional to the number of faces incident to the reflex vertex. The rest of the polyhedron is then reconstructed in time proportional to the size of the polyhedron. Thus the time complexity of the algorithm is $O(n)$, where n is the size of the SDR. Theorem 7 and the fact that the polyhedron in Fig. 9 is not uniquely reconstructible from its wire frame representation immediately lead to the following theorem.

Theorem 8. SDR *of polyhedra is a more powerful representation scheme than the wire frame for the class of polyhedra for which each face has at most* 4 *edges.*

6. Conclusion

Using the spherical dual representation, we have relaxed the requirement that the face connectivity graph of a genus 0 polyhedron must be triconnected to support unique reconstruction. All genus 0 polyhedra are uniquely reconstructible. We have also extended unique reconstruction to any polyhedron of arbitrary genus, whose SDR has degree at most 4. The completeness of arbitrary spherical dual representations remains an open question. However, as numerous attempts to construct a polyhedron of higher genus that is a counterexample have failed, we are lead to the following conjecture:

Conjecture 1. The spherical dual representation of a polyhedron of arbitrary genus is uniquely reconstructible.

This conjecture is the most intriguing question raised by this work. A resolution to this conjecture will require a deep understanding of the structure of planar-faced polyhedra of arbitrary genus. As a taste of what such understanding should entail, we offer this smaller conjecture.

Conjecture 2. No complete graph K_s, $s > 4$, is the spherical dual representation of any planar-faced polyhedron.

We also note that there is a *global* version of RECONSTRUCT that also reconstructs genus 0 polyhedra. It must consider what happens when an SDR is separated by a separation pair in the same manner as does RECONSTRUCT. We do not elaborate on this version as it is less likely to lead to a universal reconstruction algorithm for polyhedra of arbitrary genus.

We have already mentioned in the description of RECONSTRUCT how that algorithm can naturally extract many features (protrusions and depressions) of a solid. The lack of explicit order information in our approach is an advantage over the approach to feature extraction taken by Falcidieno and Giannini [3].

Another interesting aspect of our research has been the lack of the necessity for representing the adjacency information as a multigraph. In the light of our conjectures, we feel that the multigraph representation has the same power as the SDR.

An important observation is that RECONSTRUCT uses the requirement that faces are planar only to calculate $\mathcal{Q}(f)$ and the edges of P. Therefore, RECONSTRUCT actually applies to larger classes of objects than we have allowed. It would be useful to specify and investigate other classes of 'polyhedra' for which edges and the near-vertices in $\mathcal{Q}(f)$ are easy to calculate.

Acknowledgments

We wish to thank the two anonymous referees for a careful reading of the manuscript, which led to many improvements in the presentation.

References

[1] H. Edelsbrunner, Algorithms in Combinatorial Geometry (Springer, Berlin, 1987).
[2] J. Edmonds, A combinatorial representation for polyhedral surfaces, Amer. Math. Soc. Notices 7 (1960) 646.
[3] B. Falcidieno and F. Giannini, Automatic recognition and representation of shape-based features in a geometric modeling system, Comput. Vision Graph. Image Process. 48 (1989) 93–123.
[4] J.L. Gross and T.W. Tucker, Topological Graph Theory (Wiley/Interscience, New York, 1987).
[5] B. Grunbaum, Convex Polytopes (Interscience Publishers, London, 1967).
[6] P. Hanrahan, Creating volume models from edge-vertex graphs, Comput. Graph. (16) (1982) 77–84.
[7] L.S. Heath, P. Paripati and J.W. Roach, Representing Polyhedra: Faces are better than vertices, TR 92–20, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, April 1992.
[8] D. Hilbert and S. Cohn-Vossen, Geometry and the Imagination (Chelsea Publishing, New York, 1952).
[9] J.E. Hopcroft and R.E. Tarjan, Dividing a graph into triconnected components, SIAM J. Comput. 2 (1973) 135–158.

[10] B.K.P. Horn, Robot Vision (McGraw-Hill Book Company, New York, 1986).
[11] D.A. Huffman, Impossible objects as nonsense sentences, Machine Intelligence, Vol. 6 (Elsevier, New York, 1971) 295–324.
[12] D.A. Huffman, A duality concept for the analysis of polyhedral scenes, Machine Intelligence, Vol. 8 (Ellis Horwood Limited, Chichester, 1977) 475–492.
[13] J.J. Little An iterative method for reconstructing convex polyhedra from extended gaussian image, Proceedings of the National Conference on Artificial Intelligence (1983) 247–250.
[14] A.K. Mackworth, Interpreting pictures of polyhedral scenes, Artificial Intelligence 4 (1973) 121–137.
[15] M. Mantyla, A note on the modeling space of Euler operators, Comput. Vision Graph. Image Process. 26 (1984) 45–60.
[16] M. Mantyla, An Introduction to Solid Modeling (Computer Science Press, Rockville, MD, 1988).
[17] G. Markowsky and M.A. Wesley, Fleshing out wire frames, IBM J. Res. Development 24 (1980) 582–597.
[18] P. Paripati, Polyhedra: representation and recognition, M.S. Thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1989.
[19] A.A.G. Requicha, Representation of rigid solids—theory, methods and systems, ACM Comput. Surveys 12 (1980) 437–464.
[20] J.W. Roach, J. Wright, and V. Ramesh, Spherical dual images: a 3D representation method that combines dual space and Gaussian spheres, IEEE 1986 Workshop on Computer Vision: Representation and Control (1986) 236–241.
[21] J.W. Roach, P. Paripati and J.A. Wright, CAD system based on spherical dual representations, IEEE Comput. 20 (1987) 37–44.
[22] M. Sabin, Geometric modelling: fundamentals, in: J.W. ten Hagen, ed., Eurographics Tutorials '83 (Springer, Berlin, 1984) 343–389.
[23] S.A. Shafer, Shadows and Silhouettes in Computer Vision (Kluwer Academic Publishers, Dordrecht, 1985).
[24] R.E. Tarjan, Depth first search and linear graph algorithms, SIAM J. Comput. 1 (1972) 146–160.
[25] K. Weiler, Edge based data structures for solid modeling in a curved surface environment, IEEE Comput. Graph. Appl. 5 (1985) 21–40.
[26] H. Whitney, Non-separable and planar graphs, Trans. Amer. Math. Soc. 34 (1932) 339–363.